APS – Examen du 21 Mai 2008

Durée : 2 heures.

Documents autorisés

Rédiger vos réponses sur 2 copies séparées : **Copie 1 : exercice 1**
Copie 2 : exercices 2 et 3

Exercice 1 Soit $x = (x_1, \ldots, x_n) \in \mathbb{N}^n$ et $y \in \mathbb{N}$. Soient f une fonction de n variables et g une fonction de $n+2$ variables. La definition de la fonction $h : \mathbb{N}^n \to \mathbb{N}$ par :

$$\begin{array}{rcl} h(x,0) & = & f(x) \\ h(x,y+1) & = & g(x,y,h(x,y)) \end{array}$$

est primitive récursive. La fonction h est totale, c'est-a-dire définie sur toutes valeurs de son domaine, si f et g le sont.

1. Démontrer que : si f et g sont des fonctions calculables, alors h est calculable. Tenez compte dans votre preuve de la possibilité que sur certaines entrées une fonction ne soit pas définie.

2. Définir la somme, le produit et la puissance de deux entiers par récursion primitive.

3. Est-ce que la suite de Fibonacci définie par

$$F(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ F(n-1) + F(n-2) & \text{si } n > 1 \end{cases}$$

est une fonction définissable par récursion primitive ? Justifier votre reponse.

4. Construire un programme URM qui calcule la suite de Fibonacci.
Rappel des instructions disponibles :
Z(i), remise a zero du i-ème registre
S(i), incrémentation du i-ème registre
T(i,j), transfert du contenu du i-ème registre dans le j-ème registre
J(i,j,k), saut conditionnel a la k-ème instruction lorsque les i-ème et j-ème registres sont égaux.

Décrire la configuration initiale du programme et indiquer le registre où le résultat se trouvera.

Exercice 2 On considère le petit langage impératif vu en cours, dont la syntaxe est enrichie afin de pouvoir effectuer des "affectations parallèles". La syntaxe de la nouvelle construction est :

$$x, y := a_1, a_2$$

où x et y sont des variables distinctes et a_1 et a_2 sont des expressions arithmétiques. Informellement, l'exécution de cette instruction revient à évaluer les expressions a_1 et a_2 dans l'état courant et à affecter les résultats de ces évaluations aux variables x et y.

1. Définir la règle de sémantique opérationnelle permettant de décrire l'exécution d'une affectation parallèle.
2. Trouver deux expressions a_1 et a_2 et un état σ, tels que l'exécution des programmes suivants conduise à des états différents (décrire ces états).

$$x, y := a_1, a_2 \text{ et } x := a_1 \,;\, y := a_2$$

3. Définir une condition C sur l'instruction $x, y := a_1, a_2$ qui permette de garantir l'équivalence des deux programmes $x, y := a_1, a_2$ et $x := a_1$; $y := a_2$. Démontrer cette équivalence en supposant la condition C vérifiée (on pourra utiliser sans les redémontrer certains résultats vus en cours).
4. On généralise à présent la construction permettant d'effectuer des affectations parallèles :

$$x_1, \cdots, x_n := a_1, \cdots, a_n$$

   (a) Définir une condition C' sur l'instruction $x_1, \cdots, x_n := a_1, \cdots, a_n$ qui permette de garantir l'équivalence des deux programmes

$$x_1, \cdots, x_n := a_1, \cdots, a_n \text{ et } x_1 := a_1 \,;\, ... \,;\, x_n := a_n$$

   (b) Définir une condition C'' qui permette de garantir l'équivalence du programme $x_1, \cdots, x_n := a_1, \cdots, a_n$ avec le programme constitué des séquences d'affectations $x_i := a_i$ ($1 \leq i \leq n$) considérées dans un ordre quelconque.

Exercice 3 On considère dans cet exercice un petit fragment du mini langage fonctionnel implanté en TME. Les seules expressions que l'on considère sont :

x	(variables)
$\mathsf{let}\ x = e_1\ \mathsf{in}\ e_2$	(liaisons locales)
$\mathsf{fun}\ x \to e$	(définitions de fonctions)
$(e_1\ e_2)$	(applications de fonctions)

L'évaluation de ces expressions a lieu dans un environnement d'évaluation, noté E, et correspondant à une fonction qui associe une valeur $E(x)$ à chaque variable x. Les seules règles de sémantique opérationnelle que l'on considère pour les expressions sont :

$$(R_1)\frac{}{E \vdash x \leadsto E(x)} \qquad (R_2)\frac{E \vdash e_1 \leadsto v_1 \quad (x, v_1), E \vdash e_2 \leadsto v_2}{E \vdash \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2 \leadsto v_2}$$

$$(R_3)\frac{}{E \vdash \mathsf{fun}\ x \to e \leadsto << x, e, E >>}$$

$$(R_4)\frac{E \vdash e_1 \leadsto << x, e_f, E_f >> \quad E \vdash e_2 \leadsto v_2 \quad (x, v_2), E_f \vdash e_f \leadsto v}{E \vdash (e_1\ e_2) \leadsto v}$$

1. On se place dans un environnement d'évaluation E quelconque. Montrer que les deux expressions :

$$((\mathsf{fun}\ x \to e_2) \quad e_1) \quad \text{et} \quad \mathsf{let}\ x = e_1\ \mathsf{in}\ e_2$$

   s'évaluent à la même valeur dans E.
2. Montrer (par induction sur e) que toute expression e s'évalue en une valeur :

$$\forall e\ \forall E\ \exists v\ E \vdash e \leadsto v$$